

# Commodore Users Group of Saskatchewan

January, 1990

Vol 5. No. 1

**M  
O  
N  
I  
T  
O  
R**

## Table of Contents

Obligatory Stuff	1	Editorial	1
President's Message	2	Agenda	2
128 Windows	3	Experts List	4
Meeting Dates	5	Learning, Decline & Fall	5
Scratch and Save	6	New Club Disks	6



*Happy  
New Year*

The Monitor would like to extend its wishes for a safe, happy and prosperous New Year for the members of CUGS.



# Obligatory Stuff



Editorial  
by  
Jarrett Currie

## CUGS MAILING ADDRESS:

CUGS  
143 Birchwood Cres.  
Regina, Sask.  
S4S 5S3

CUGS BBS - (386) 586-1189

President	Barry Bircher	359 1925
Vice President	Richard Maze	586 3291
Treasurer	Real Charron	545 7601
Editor	Jarrett Currie	757 2391
Asst Editor	Shaun Hase	584 3371
Librarian	Keith Kasha	359 1748
Asst Librarian	Steve Boques	949 1370
Members at Large	Ken Danylczyk	545 0644
	Harry Chong	789 2142
	Earl Brown	543 2068
	Gord Williams	543 0373
	Joe Gomes	789 8174

If you have any questions about CUGS please feel free to contact any of the above executive members.

THE MONITOR is published monthly by the COMMODORE USERS' GROUP OF SASKATCHEWAN (CUGS), Regina, Sask., Canada. CUGS meetings are held at 7 pm the FIRST WEDNESDAY of every month (unless otherwise noted) in the North-West Leisure Centre, corner of Rochdale Boulevard and Amason Street.

Anyone interested in computing, especially on the C64, 128 or 64C, is welcome to attend any meeting. Out of town members are also welcome, but may be charged a small (\$3.00) mailing fee for newsletters. Members are encouraged to submit public domain software for inclusion in the CUGS DISK LIBRARY. These programs are made available to members. Any member is entitled to purchase DISKS from our public domain library for a nominal fee. Programs are 'freeware', from computer magazines, or the public domain. Individual members are responsible for deleting any program that he/she is not entitled to by law (you must be the owner of the magazine in which a particular program was printed). To the best of our knowledge, all such programs are identified in their listings. Please let us know if you find otherwise. Contact our club Librarian, Keith Kasha.

CUGS is a non-profit organization comprised of C64, 64C, C128, and 128D users interested in sharing ideas, programs, knowledge, problems and solutions with each other. The more members participate, the better the variety of benefits. Membership dues are pro-rated, based on a January to December year.

Happy New Year! I hope you all have a prosperous and rewarding year.

I would feel amiss if I didn't participate in welcoming aboard the new club executives. Especially, I would like to thank Shaun Hase for volunteering his time in helping with the Monitor. I know that Shaun has a keen interest in programming and graphics, and hopefully he will share those talents with the rest of us.

Because, it doesn't seem that there are that many programmers around these days. Lately, whenever I meet someone who has purchased a computer, I invariably hear that they didn't buy the computer to program it, nor did they even intend to learn! Many people share the following beliefs:

- I can't learn to program because I don't do math that well.
- These things are way too complicated. I'd rather buy a simple program.
- I only wanted to do word processing with it, and I already have the word processor bought.

If you hold one of these beliefs, for shame! When you sit at your keyboard, you have at your instant disposal a machine that you can command in a way that is only limited by your imagination. If you learn to program it, you will be able to create a program that handles YOUR finances, in exactly the way you way it to. You will be able to create graphics and animation that is not limited in any way by the author of any program. You will be able to write programs that manipulate your files and programs in any way that you dream possible. You will be able to write programs that will substitute for human players when a game partner is not available.

But, more than that, you will have fun.

We have several programmers in our midst, myself included. I can only imagine that after the combined years of programming that we share, if we put all of our finished, useful programs on disk, we would have many free blocks left! But, the hours we have spent.

For what? you may ask. For the sheer pleasure of seeing if we can do it. For seeing if we can add every imaginable feature to the most bland programs. To see if perverting the language beyond recognition can achieve anything useful. To see how much we can learn about our machines while we program.

And it really isn't that hard. Apart from what you have heard, programming has very little to do with math. As a professional applications programmer, my day involves the most simple additions and subtractions, with just a few

multiplications thrown in for variety. When you realize that computers see only three number - 0, 1, or 2 - it would be hard to imagine how it could come up with a difficult calculation. No, the difficulty isn't in the math. The true challenge is making the plunge into programming. When you finally decide that the next SYNTAX ERROR you get will not go uninvestigated, you will have met the most difficult challenge of computing.

And you will be richer for it. You will develop skills that will not only help you use your computer more successfully, but you will also be entertained. All of this sitting on top of your desk!

If my article has even slightly piqued your interest in programming, don't hesitate to ask any of the club members how to get started in programming. And when your first programming creation is finally finished, don't hesitate to put it on a club disk!

Have a wonderful decade.



## PRESIDENTS MESSAGE

Hello everyone. Glad to see you made it out for the meeting, if not, then hope to see you out at one real soon. This is my first installment as the new President of the club. I believe it is appropriate at this time to say farewell to the old executive and hello to the new. I have been with this club since December of 1985. Since that time I have seen Richard Maze perform the duties as President as if it were second nature. He has done an excellent job of leading the club. I tip my hat to him as he steps down to Vice President. (An EASY job. I know, I had it!) It is my sincere hope to be able to fill his shoes and perform his duties in a similar manner. However, I am sure to crash things at the start as Rome was not built in a day.

One of the things I have noticed lately is the stigma that is put on club executive positions. It seems that club executives are labelled as higher-ups or gurus of the computer world. Nothing can be further from the truth. An executive is nothing more than a regular member with a few additional responsibilities. Every member, regardless of his/her responsibility, is required to put in as much as is taken out. It is my hope to see members of this club get more involved in what goes on and help out wherever they can. I don't mean for you to run for a position, it is not necessary. All that is needed is for everyone to put some things INTO the club as well as get things OUT. If all one does is take..take..take and no give, the club will go stale and go flat. If you have a program or game that you like (or DISLIKE), please write a few lines, paragraphs, pages or whatever and give it (either on paper or on a disk (returned)) to one of the clubs executives or upload it to our own BBS. If you have a program that is unusual or

even usual, bring it or upload it to the club and let others have a chance to look at it. If YOU don't, then WHO will. The more input from every one the better this club will be.

I am proud to be a computer user, and I want to know more about it. One of the reasons for joining the club has been the fellowship with other users. To exchange ideas, programs, methods and programming tricks. Every time I attend a meeting I find out more new things (at least one anyway) that was not well known before. For example, in writing a program you are continually modifying and improving it. In the process of scratching and saving, one is able to inadvertently misname or scratch the wrong file. A very easy way of avoiding all the typing and thinking (read lazy) about the syntax of the commands and so forth, is to type the commands after a REM statement at the beginning of your program once. At the time you need to save, scratch or whatever, all you do is cursor up to the line you entered and space out the line number and REM and hit return. VOILA. Spacing out or erasing the line number and REM statement and hitting return automatically is interpreted as an immediate command and is executed as such. This one trick saved me countless keystrokes and allowed me to concentrate on the next problem:

```
10 REM my special money maker program
20 REM open15,8,15,"R0:money maker bu=money
   maker":close 15
30 REM open15,8,15,"S0:money maker":close15:save"money
   maker",8
40 your program starts here_____
```

Well, until next time. There is nothing as constant as change.

January CUGS meeting

Wednesday, January 3, 1990  
Northwest Leisure Centre

### Agenda

Beginner's Session (Computer setup)  
File managing on the 64  
File managing in the 128 DOS shell

Break

Door prize draw  
Trouble shooting questions/answers



# 128 Windows

by  
Shaun Hase

At the last meeting, Richard had been talking about teaching his classes how to create their own character sets. This brought back some memories of my own, sitting at the VIC with graph paper, creating unique character sets for some particular program. When I first got my 128 I thought that character set generation would be as simple as on the VIC. Then I found out about banks. The 128 has 16 of them. I have yet to understand fully how each of these memory configurations can be put to use, but they were a major stumbling block. I dug through some magazines and reference books, figured something out and, subsequently, wrote the following program.

```
100 TRAP390
110 SCMLR:BANK14:CM=53248:RA=8192
120 GRAPHIC1,IFAST
130 DEFFNP(R)=PEEK(CM+R)
140 FORR=0TO1023
150 : POKERA+R,FNP(R)
160 NEXTR
170 FORT=1024TO2047STEP8
180 : FORS=TTOT+6
190 : POKERA+S,255-FNP(S)
200 : NEXTS
210 : POKERA+S,FNP(S)
220 NEXTT
230 BANK15
240 POKE2604,24POKE217,4
250 GRAPHIC0:SLOW
260 PRINT
270 PRINT"CHARACTER SET GENERATED."
280 PRINT"SAVE TO DISK AS BINARY FILE?"
290 DO
300 : GETKEYA$
310 LOOPWHILEA$<"Y"ANDAS<"N"
320 FA$="Y"THENBEGIN
330 : INPUT"FILENAME";FI$
340 : FI$=LEFT$(FI$,12)+" ".BIN"
350 : BSAUE(FI$),B15,P8192TOP10240
360 : PRINT"FILE SAVED AS: ";FI$
370 BEND
380 IFER>0THENBEGIN
390 : SLOW:GRAPHIC0
400 : PRINT"AN ERROR HAS OCCURRED."
410 : PRINT"BASIC ERROR: "ERR$(ER)
420 : PRINT"LINE NUMBER:"EL
430 BEND
440 END
```

This program creates an underlined character set for the 128 in 40-column mode. Instead of the cursor being a solid square, it now becomes a line. In the development of the

program and in the way the 128 manipulates character data, the reversed character set is lost. What it is replaced with is an underlined character set. Also, for ease of programming, the 40-column graphics screen is disabled and used for character storage. You can never get something out of nothing. Hopefully, some of you will find the program useful.

In the 128, character memory, at least the 40-column set, is stored in locations 53248 to 57343, in bank 14. The only way to manipulate the character set is to move a copy of the memory from ROM into RAM, preferably somewhere where it will not be corrupted. Once the memory is copied into RAM, it can be changed in any way desired and by changing the values of two memory locations, the 128 can be told to look for character memory where the new character set is. After doing so, a switch to either bank 0 or 15 (15 seems to be a favourite of programmers), is required to gain full access of 128's capabilities. A quick look at a memory map in bank 0 or 15, as in most reference books, show that location 53248 is the beginning of VIC chip registers, not character ROM. This bank switching, however confusing it might be, gives the 128 a substantial amount of power, if it is done properly.



One more thing to know about the program before a full explanation is done is to realize that only one character set (256 characters) can be displayed on the 40-column screen at one time, so the starting position of the redefined character set can take on two values: 53248 and 55296. 53248 - 55295 is uppercase/graphics and 55296 - 57343 is lower/uppercase. Either set can be redefined, but only one set can be used at one time.

Through the refinement of the program, some BASIC 7.0 commands have been implemented, as well as functions. Although it is questionable to whether the use of functions speed up BASIC, it gives for a neater program. With the BASIC 7.0 commands, GOTO's were eliminated and an error trapping routine was added. The error trapping routine was more of a necessity than anything else. When FAST is executed, the 40-column screen is disabled and if an error occurs, there is no way of knowing that.

The key lines to the program are 110 to 250. Bank 14 is selected, some key variables are set, and the GRAPHIC statement is issued to raise the start of BASIC from 7168 to 16384. Lines 140-160 do a simple byte-for-byte copy of the first half of the character set or the first 128 characters. Lines 170-220 do sort of the same thing for the next 128 characters, but as this is the reversed character set it is reading, and little mathematical logic is

needed to convert the data. Since only the first seven bytes of each character are to be changed (a character is made up of eight bytes), lines 180-200 accomplish this. The original character byte is read and is subtracted from 255 to obtain the new byte. Line 210 reads the last character byte and stores it with no manipulation, thus producing an underline effect.

When all the characters are read and converted, bank 15 is selected and two memory locations are changed so the 128 will look to the proper place for character data. Location 2604 is the VIC text screen/character base pointer. Since we want the 128 to read character data starting at location 8192, this location must have the correct value. The memory location must be an even multiple of 1024. So, divide the starting address by 1024 and add 16. I looked at two references, listed below, and the above method seemed much easier for reprogramming the pointer. The second location, 217, is the RAM/ROM VIC character fetch flag. Normally, this byte holds the number zero, which tells the 128 to look for character memory in ROM. A value of four (4) tells the 128 to look for character memory in RAM.

The remainder of the program serves two purposes. One, is to trap errors that could occur during FAST mode. This was left in as an example of how to trap errors while working with the 40-column screen in FAST mode. Two, is that once the character set has been created, the program allows for you to save it as a binary file. This enables you to reload the character set quickly and easily without having to run the generator program again. To load in the character set, enter the following:

```
10 GRAPHIC1:GRAPHIC0
20 BLOAD"CHARSET.BIN",B0,P8192
30 POKE2604,24:POKE217,4
```

Also, if you have inadvertently pressed RUN/STOP RESTORE and want the character set back, typing the following line in immediate mode should do the trick.

```
POKE2604,24:POKE217,4
```

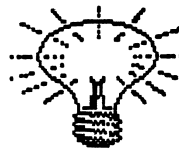
I hope some of you find this program useful, if not for the underline character set itself, but for possible reprogramming of certain characters.

#### References

Brannon, Charles. "Ultrafont+." COMPUTE!'s Gazette, 39 September 1986, 48.

Commodore Business Machines, Inc. "The Power Behind Commodore 128 Graphics." Chap. in Commodore 128 Programmer's Reference Guide. Toronto: Bantam Books, 1986.

Commodore Business Machines, Inc. "The Commodore 128 and Commodore 64 Memory Maps." Chap. in Commodore 128 Programmer's Reference Guide. Toronto: Bantam Books, 1986.



## Experts List

Recently, we began a regular service to our membership. The people below have agreed to let their names be listed as "experts" in some aspect of C64/128 computing. If you've a question, these brave volunteers can likely answer it, or help you find an answer that works. If you have a skill at some computing process, consider listing yourself with our other volunteers. We're all in this together!

#### Wordprocessing

Paperclip III	Shaun Hase	584 3371
Paperclip (to version E)	Richard Maze	586 3291
Paperclip (to version E)	Jarrett Currie	757 2391
Paperclip (any version)	Ken Danyleczuk	545 0644

#### Spreadsheets

Multiplan	Richard Maze	586 3291
Pocket Planner	Barry Bircher	359 1925
Better Working SS	Ken Danyleczuk	545 0644

#### Databases

Pocket Filer	Barry Bircher	359 1925
Oracle (Consultant)	Ken Danyleczuk	545 0644

#### Communication

Pro128Term	Barry Bircher	359 1925
Pro128Term	Jarrett Currie	757 2391
Library files	Barry Bircher	359 1925
Desterm 128	Gilles Archer	545 4638

#### Music/Sound

(Most)	Ken Danyleczuk	545 0644
--------	----------------	----------

#### Languages

Forth	Ken Danyleczuk	545 0644
Pascal	Ken Danyleczuk	545 0644
ML (machine language)	Ken Danyleczuk	545 0644
ML (machine language)	Barry Bircher	359 1925
BASIC (general)	Richard Maze	586 3291
BASIC 7.0 (graphics)	Shaun Hase	584 3371
BASIC (2.0-7.0, files)	Ken Danyleczuk	545 0644

#### Graphics

Print Shop/Master	Ken Danyleczuk	545 0644
Koala Painter/Printer	Ken Danyleczuk	545 0644

#### Hardware

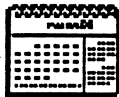
All hardware	Tyler Rosewood	525 0214
Disk Drive Maintenance	Ken Danyleczuk	545 0644

#### GEOS

GEOS 64 and 128	Tyler Rosewood	525 0214
GEOS 64	Jarrett Currie	757 2391

## CUGS Meetings for 1990

The meeting dates for the first half of 1990 from January to May have been established and are held on the first Wednesday of the month in room #2 at the Northwest Leisure centre and start at 7.00 P.M. Please mark these dates down and make plans to attend on these dates. The following are the dates for Jan - May 1990:



Wednesday, January 3, 1990  
Wednesday, February 7, 1990  
Wednesday, March 7, 1990  
Wednesday, April 4, 1990  
Wednesday, May 2, 1990



## Learning, The Decline and Fall

by  
Ken Danylozuk

This year will mark the NINTH year of use for Commodore C64's (6 for the 28!). Well over 9 million combined sales, mostly in the home market. That's a lot of machine owners, but how many users are there, really?

The answer is probably a multiple set of percentages depending on your definition of "user". I'd estimate about 90% are game playing users, 50% are simple home utility (word processing, databasing, etc.) users, maybe 25% are intense multi-application users with maybe 10% real in-depth users, capable of manipulating the machine through a variety of languages, programming and customizing programs. Our club began with that 10-25% group and for a few years grew with that in-depth type of user. That type actually spent the money to own a machine, and owning a machine was a consuming passion for them.

Of late, however, the past three years or so, the "surface" user has come in to the majority - this is the user who would LIKE to do a lot with his/her computer but doesn't want to invest a great deal of time in learning the inner workings of the machine before he can do what he wants. This has brought a dilemma of the first order - because programmers have developed "friendlier" and simpler screen interfaces to ease the "learning curve" for the masses. The drive to separate the effort of learning from the use of the computer as a tool has led to many fascinating innovations - most notably GEOS, a whole new operating system.

Unfortunately, it has also developed a backwards attitude towards personal effort to achieve desired results. English translation: it is incredible how lazy one can become in order to achieve an end. And the attitude is being nurtured by all the new technologies. Buy a VCR that virtually tunes itself, makes programming for recording a tedious strip of button pushing, electronic analyzers in car electrical and mechanical systems that tell us when and where to take the car for service (God forbid you might do the servicing yourself!). Programs from simple games to complex utilities

that can be operated casually WITHOUT READING THE INSTRUCTIONS!

That's my point - software you can use without reading the instructions - except the menus offered can't possibly cover every situation, so they opt for the "most used" functions of the program. This means that often a feature of a program that one might find useful lies unknown or unused because it "isn't on a menu". Often the software gets panned or gets a less than enthusiastic endorsement because no one read the instructions! This is a problem with everything from games to utility programs, and is beginning to reach epidemic proportions in some situations.

So what put my shirt in a knot? I'm on of the "experts" in the "experts list" found in the Monitor, and I really don't want to sound like I'm complaining, because I like helping people; I enjoy the calls and conversations with fellow computerists, and I like to spread the 8-bit gospel wherever I can. I've found that most of my requests for help involve people learning something that was explained (clearly, even) in the manual. Sure, it's easier to learn if someone shows you, and a lot more fun usually. But sometimes some comments after the teaching session have disturbed me.

On one occasion a month ago I was asked about sorting a list using Paperclip, one terrific word processor. Actually, my involvement came after a disturbing series of events. The young lady involved had gone with her problem to a local computer shop, to be told by the youngish clerk that for what she wanted to do, she required a proper database program, and proceeded to sell her PocketFiler. Now, it's a good program, too, but it didn't allow her to do what she wanted any more easily. In desperation she tried another (friendlier) shop who suggested she try our club if she was using a C64, and my name was used. A 20 minute telephone conversation made her situation and needs quite clear. My advice was that she: 1) return PocketFiler, it really wasn't necessary; 2) get her manual and learn to use the index (at the back); 3) call me back after she'd read three sections we discussed. She did all three, and our next conversation was 2 minutes long, to clarify a poorly written section in one chapter. Presto! One sorted mailing list, and the capability to print personalized labels and form letters.

But her closing comments bothered me. To paraphrase, she said she didn't realize that using the C64 was so complicated - she wondered out loud if she couldn't have done all this more easily with an IB\* or a M\*cl\*toosh. More easily. The words haunted me for the weekend. Easier than what - reading 12 pages and learning 6 new keystroke commands for a wordprocessing program you'll use for many years yet? Easier?

I can't claim innocence in this mistaken point of view. I've been lulled into the "ease of use" syndrome a few times myself. Just last week my wife requested that I help her by typing a batch of letters and a special certificate for a family event. I've become quite comfortable using an MS-DOS machine with a speedy hard drive and sleek, quick

(read "expensive") programs that would make the job a snap, but... I'm on holidays and that machine's at the office. I almost said she'd have to wait until I could get down to the office, when I caught myself in mid-sentence. Wait for what? I've used Paperclip and GEOS for small projects for the MONITOR, because I believe in the quality and ability of the software - it IS powerful and it does a GOOD job! I was a little rusty with the special features of both, having been lulled in the cradle of "Big Blue" for the last while.

So all this leads to one serious piece of advice - take those instruction books and operating manuals out of the cupboard, packing boxes or off the shelf and open 'em up. You might be surprised at what five minutes reading will provide. You might find brand new features in an old favourite piece of software.

One added piece of advice - keep your manual handy and learn to use the index, even for the most menu driven program. Use that index and our own club experts (just a phone call away) before you claim it can't be done.



## Scratch and Save

by  
Earl Brown  
(Member at large)

The library received a modified version of the CUG's loader so that it will work in both the 40- and 80-column mode of the 128 computer. The conversion is the work of Richard Maze (who doesn't have a 128) and Barry Bircher (who has). The program can be downloaded from the bulletin board, and if you buy any of the new programs in the 128 library, it will also be on them.

In order for the program to work in both the 40 and 80 column modes of the 128, the scrolling message incorporated in the 64 loader has been deleted. Also, the Function Keys have been changed to: R - Load and run, D - Description, L - Load only and N - Next page. If you do not reassign the 128 Function Keys, F-5, when pressed, can also be used to display the description line.

As with the 64 version, you have to prepare a sequential file containing the name of the file to be loaded on the first line, the word "RUN", "SYSxxxxx" or "LIST" on the second line, and a description of the program on the third and fourth lines. You repeat this procedure for all the programs you have saved to the disk. All the clubs' disks have less than 664 blocks saved on them so that they can be used on either a 1541 or a 1571 disk drive.

A final and important bit of information about the loader. Like the 64 loader, all the programs that are written in basic or with a basic boot program that require the word "RUN" to execute them, must be located at the 128 BASIC start location 7169 (for the 64, 2049). So if you have a program written for the 128 but it was actually saved on a 64, pet, plus 4, 16, or a VIC, its starting location will be different, and instead of loading the program required, the computer will execute the CUG's loader again. If this has

happened turn your computer off and on, DLOAD the program, scratch the original from disk, and rescue the program you now have in the computer memory under the same name. Now you can [SHIFT/RUN] the CUG's loader, cursor to the file in question and press "R" again. This time your program should load and run.

If you examine the library listings, you'll notice two disks ready for the 128 in their new format: 128 ARCADE GAMES #ZA and 128 ARCADE GAMES #ZB

Next month, there will be more. Also you will find the CUG's January 1990 Gazette disk and the July/August 1989 Run disk.

## New Club Disks

CUGS GAZETTE JAN #90

fastloader/1581  
bitmap effects  
pointer demo  
pointer setup  
flags  
royal rescue  
error trapper  
geoconverter 13  
text scroller  
file scanner

CUGS JUL/AUG 89 #80

menu 128 (RUN)  
menu 64 (RUN)  
run shell  
a-z scramble  
cubix 1/128  
mousetrap 64  
mousetrap 128  
label base  
memory tiles  
thoroughbird/128  
ketchup attack  
ketchup.49152  
run's checksum

128 ARCADE GAMES #ZA

litter patrol  
deepsix/128  
letter game  
mountain miner  
rotation tag  
lexitron/128  
pig\$forbucks/128  
moon rescue/128  
auto race/128  
connect 'em/128  
tile trader/128  
skidders/128  
chess tour/128  
backgammon v3.7  
alhedron  
casino128  
wheel/fortune  
puzzle making  
depth charge

128 ARCADE GAMES #ZB

galactic cab co.  
gumball rally  
chess/128  
cribbage/128  
block out/128  
gridlock/128  
bacteria/128  
math magic/128  
monster bbq/128  
wraiths/128  
galaxy cabco/128  
archer/128  
chainmail/128  
empire/128  
thoroughbird/128